

Eigener DynDNS Dienst

Eine eigene Toplevel Domain die direkt immer auf die IP des Internetanschlusses zuhause zeigt. Quasi der eigene DynDNS-Dienst – ohne die Hilfe von (kostenlosen oder kostenpflichtigen) DynDNS-Providern. Das wärs!



Für dieses Setup braucht man:

- **Eine eigene Domain** (am einfachsten über [Hostpoint](#) registriert)
- **Einen Slave-DNS Eintrag bei den Hostpoint DNS Servern**, (der auf meine aktuelle IP zeigt)
- **Einen Netzwerk lokalen root-DNS-Server** der alles managed (meinem Fall Ubuntu 16.04)

Einleitung

Für jede Domain (die bei Hostpoint registriert wird) braucht man **zwei Namensserver** welche sagen können welche IP hinter der Domain steckt. Diese Server sind die autoritiven Nameserver. Laut den Vorgaben von Hostpoint, sollten hier die eingetragenen Nameserver *nicht* genau die gleiche IP-Adresse haben und auch nicht ähnliche Adressen (gleicher Netzbereich).

Da wir nun, unserem root-Server hierfür nicht extra zwei verschiedene IP-Adressen einrichten wollen, nur damit Hostpoint zufrieden ist, brauchen wir also noch einen zweiten Nameserver der nichts anderes macht als die Daten unseres Nameservers zu übernehmen. Dies ist der sogenannte **Slave-Nameserver** und man kann diesen bei Hostpoint einfach per Klick erstellen (bzw. einen Eintrag auf dem Nameserver vornehmen.).

Für diese Anleitung gehe ich davon aus dass ein frisch installiertes **Ubuntu 16.04** als späterer **root-Server meinroot.ch** zur Verfügung steht und die **Domain blackgate.org** mir gehört.

Installation

Zuallererst wird auf meinroot.ch den DNS-Server installiert:

```
# apt-get install bind9
```

Als nächstes, wird in Bind eine neue Zonendatei für blackgate.org erstellt, diese liegt in /etc/bind/db.blackgate.org und sieht wie folgt aus:

```
$ORIGIN .
$TTL 60; 1 minute
blackgate.org      IN SOA  www.meinroot.ch. dn.meinroot.ch. (
                    2017033101; serial
                    7200; refresh (2 hours)
                    1800; retry (30 minutes)
                    604800; expire (1 week)
                    21600; minimum (6 hours)
                    )
                    NS   ns1.first-ns.de.
                    NS   www.meinroot.ch.
                    NS   robotns2.second-ns.ch.
                    A    89.14.9.131
                    MX   10 mail.blackgate.org.blackgate.org.
                    AAAA 2002:590e:2e2e:0:ele3:60f5:97d:9429

$ORIGIN blackgate.org.
mail      CNAME  blackgate.org.
www       CNAME  blackgate.org.
```

Jetzt erzeugen wir einen Update-Key, damit unser Server die Datei später auch automatisch updaten darf:

```
# /usr/sbin/ddns-confgen -z blackgate.org
```

Diesen Key schreiben wir in die Keydatei `"/etc/bind/ddns-key.blackgate.org.conf"`:

```
key "ddns-key.blackgate.org" {
    algorithm hmac-sha256;
    secret "CTQpYK6Dfdfs334fsd08LSY7ApziY3423fds8zw=";
};
```

Das Key-File müssen wir anschliessend in der bind-Config `"/etc/bind/named.conf"` inkludieren:

```
// Update-Keyfile
include "/etc/bind/ddns-key.blackgate.org.conf";
```

Dannach wird der Key auch noch in `"/etc/bind/named.conf.local"` hinterlegt:

```
zone "blackgate.org" {
    type master;
    file "/etc/bind/db.blackgate.org";
    update-policy { grant ddns-key.blackgate.org zonesub ANY; };
};
```

Nun muss noch erlaubt werden, dass die Hostpoint-Nameserver überhaupt einen Zonentransfer machen dürfen. Dafür, wird in der Datei `"/etc/bind/named.conf.options"` folgendes eintragen:



```
// Zonen-Transfer fuer Hostpoint Nameserver
allow-transfer { 213.239.242.238;213.133.105.6;193.47.99.3; };
```

Jetzt brauchen wir noch ein Script, welches dann anschliessend von aussen das Updaten des DynDNS-Eintrages vornimmt - und zusätzlich auch gleich noch zu Log-Zwecken einen Eintrag in eine MySQL-Tabelle durchführt:

Dafür gibt es folgendes Skript, welches hier → `/root/bin/update_dyndns.sh` abgelegt wird:

[/root/bin/update_dyndns.sh](#)

```
#!/bin/bash

NS="127.0.0.1"
NSKEY="/etc/bind/ddns-key.blackgate.org.conf"
TMPFILE="/var/www/html/dyndns.txt"
ZONE="blackgate.org"
TIMEOUT=60

OLDIP=$(dig @$NS $ZONE A | grep ^$ZONE.*A | awk '{ print $5 }')
NEWIP=$(cat $TMPFILE)

echo "old: $OLDIP"
echo "new: $NEWIP"

if [ "$OLDIP" != "$NEWIP" ]; then
    echo -e "
update delete $ZONE. A
update add $ZONE. $TIMEOUT A $NEWIP
show
send" | /usr/bin/nsupdate -k $NSKEY
rndc freeze $ZONE
rndc thaw $ZONE
else
    echo "no update required"
fi
```

Dieses Script holt die aktuelle IP aus der **Datei** `/var/www/html/dyndns.txt` und kümmert sich sowohl um das Update des Nameservers, als auch ums weiterleiten der Zonendaten an den Hostpoint-Slave Nameserver.

Diese Shell-Datei, wird dann automatisiert von Zeit zu Zeit, per [cronjob](#) von root ausgeführt. Zum erstellen, des cronjobs, wird die `/etc/crontab` Datei editiert und folgender Eintrag eingefügt:

```
* * * * * /root/bin/update_dyndns.sh > /dev/null 2>&1
```

Jetzt wird nur noch ein PHP-Skript gebraucht, welches per URL erreichbar ist und die **Datei** `/var/www/html/dyndns.txt` mit der übergebenen IP beschreibt. Dazu, wird folgendes PHP Skript gebraucht:

[/var/www/html/IP_updater.php](#)

```
<?php
//
$username = "meintollerdyndnsdienst";
$password = "miteinemgeheimenpassword";
$dyntxt = "/var/www/html/dyndns.txt";
$db = "dyndns0ettingen";

if($username == $_GET['username'] && $password == $_GET['password'])
{
    $a = fopen("$dyntxt", "w");
    fwrite($a, $_GET['ipv4']);
    fclose($a);

    $mysqli = new mysqli("localhost", $username, $password, $db);
    if ($mysqli->connect_errno)
    {
        echo "Failed to connect to MySQL: (" .
        $mysqli->connect_errno . ") " . $mysqli->connect_error;
    }

    if (!$mysqli->query("INSERT INTO dyndns(domain, ipv4, ipv6)
VALUES ('" . $_GET['domain'] . "', '" . $_GET['ipv4'] . "', '" .
$_GET['ipv6'] . "')"))
    {
        echo "Insert-Error: (" . $mysqli->errno . ") " .
        $mysqli->error;
    }

    mysqli_close($mysqli);

    echo "success";
}
?>
```

Username und Passwort werden nicht nur im Script geprüft um zu checken ob derjenige den Update machen darf, sie werden auch gleich zum anmelden an Mysql genutzt. Es muss also der entsprechende Mysql-Nutzer existieren. Das Tabellenformat ist einfach:

```
CREATE TABLE `dyndns` (
  `updatetime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `domain` varchar(256) CHARACTER SET utf8 NOT NULL,
  `ipv4` varchar(25) CHARACTER SET utf8 NOT NULL,
  `ipv6` varchar(100) CHARACTER SET utf8 NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Nun müssen wir bei Hetzner einen Slave-DNS Eintrag bestellen, dort tragen wir die IP von meinroot.ch ein.

Bei der Denic hinterlegen wir nun als Nameserver 1 unseren meinroot.ch und als Nameserver 2 irgendeinen der Hetzner-Nameserver (ns1.first-ns.de, robotns2.second-ns.de, robotns3.second-ns.com)

Nun sollten wir alles schon gut mit dem Aufruf der URL testen und debuggen können. Zum prüfen was die Nameserver gerade als IP für blackgate.org rausgeben dienen folgende Befehle:

```
# dig @127.0.0.1 blackgate.org
# dig @www.meinroot.ch blackgate.org
# dig @ns1.first-ns.de blackgate.org
# dig @robotns2.second-ns.de blackgate.org
```

Die Fritzbox bekommt nun folgende Konfiguration unter DynDNS:

```
Dynamic DNS-Anbieter: Benutzerdefiniert
Update-URL:
https://www.meinroot.ch/dyndns.php?username=<username>&password=<pass>&domain=<domain>&ipv4=<ipaddr>&ipv6=<ip6addr>
Domainname: blackgate.org
Benutzername: meintollerdyndnsdienst
Kennwort: miteinemgeheimenpasswort
```

Meine Fritzbox brachte hier noch im Systemlog alle paar Minuten folgenden (nicht tragischen) Fehler:

Dynamic DNS-Fehler: Die Dynamic DNS-Aktualisierung war erfolgreich, anschließend trat jedoch ein Fehler bei der DNS-Auflösung auf.

Wenns einen ströt wird man das los indem man sich mit Telnet auf die Fritzbox verbindet und die Datei ‚var/flash/ar7.cfg‘ so verändert dass under ‚liveday‘ ‚1d‘ steht.

-
- <https://www.thomaschristlieb.de/eigener-dyndns-dienst-mit-fritzbox-und-hetzner/>

Last update: **2017/03/31 12:37**